**1 METODOLOGIA**

Para o desenvolvimento do trabalho a abordagem orientada a objetos será empregada, para o desenvolvimento do sistema web. Os objetos do sistema estão divididos em estereótipos que são: *Model*, *Controle* *e Apresentação*. Para o modelo gerado será utilizado a linguagem UML por meio dos diagramas de Casos de Uso, Classes, Sequência.

Sistemas orientados a objetos visam definir classes de objetos e seus relacionamentos que modelam o sistema. Esses sistemas são mais fáceis de mudar quando comparados à abordagens estruturadas. Os objetos contém dados e operações para serem manipulados *(Engenharia de Software, SOMMERVILLE, 2011, ED. 9ª).*

A UML é uma linguagem padrão para modelagem orientada a objetos. O objetivo da UML é fornecer múltiplas visões do sistema sob variados aspectos por meio de diagramas divididos em estruturais e comportamentais. Desta maneira é possível atingir a completude da modelagem *(UML 2, GUEDES, 2011).*

**1.1 REQUISITOS DE SOFTWARE**

Os requisitos de software são classificados com requisitos funcionais (RF) e requisitos não funcionais (RNF). Os requisitos funcionais são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas. Os requisitos não funcionais são considerados restrições ao sistema tanto para funções como para serviços. Os requisitos que são aplicados a todo o sistema são conhecidos neste trabalho como requisitos gerais (RG), porém eles podem ser diretamente vinculados a determinado requisitos funcionais.

Ao longo do documento os requisitos serão apresentados de forma simples por meio de tabelas que apresentam a descrição do requisito funcional e os requisitos não funcionais associados ao requisito descrito. A Tabela 1 apresenta o modelo de tabela para os requisitos funcionais. Os requisitos serão incrementados de um em um e os não funcionais associados recebem o número do funcional e é adicionado uma segunda numeração, para exemplificar o primeiro requisito será o RF1 e os requisitos associados são RNF1.1 até RNF1.x.

**Tabela 1: Requisito Funcional Modelo de Requisito Funcional**

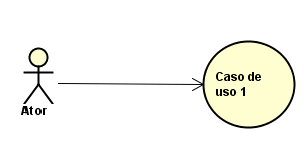
|  |  |
| --- | --- |
| **RF1 – Requisito Funcional Modelo de Requisito Funcional** | |
| **Descrição:** Descrição do requisito funcional. | |
| **Requisitos Não-Funcionais Associados** | |
| RNF 1.x | Requisito 1.1 não funcional. |

Fonte: Grupo CRPO(2016)

**1.2 CASOS DE USO**

O diagrama de casos de uso apresenta uma visão em linguagem simples do comportamento externo do sistema, ou seja, apresenta as funcionalidades *(UML 2, GUEDES, 2011).*

Este diagrama é composto por dois itens principais, sendo estes os atores e casos de uso. Os atores representam os papéis dos usuários que interagem com o sistema, são representados pelo desenho de um boneco. Os casos de uso referem-se a serviços, tarefas ou funcionalidades e que podem ser utilizados pelos atores do sistema *(UML 2, GUEDES, 2011).* A Figura 1 apresenta os dois componentes de um diagrama de casos de uso.

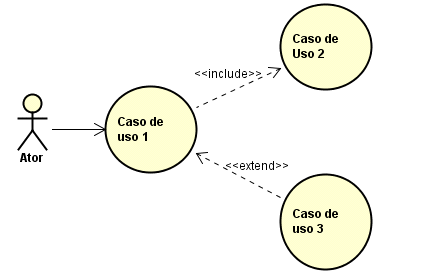


**Figura 1: Exemplo de Caso de Uso**

Fonte: Grupo CRPO(2016)

O diagrama de casos de uso possui associações que podem ser estabelecidas entre atores, atores e casos de uso ou entre casos de uso. Os relacionamentos entre casos de uso são classificados como inclusão (*include*), extensão (*extends*) e generalização/especialização.

A generalização faz com que um caso de uso herde o comportamento e associações de outro caso de uso. O desenho da generalização é uma seta apontando para o caso de uso geral e a outra ponta do traço é ligada aos casos de uso especialistas. A inclusão é utilizada quando dois ou mais casos de uso possuem uma situação em comum, obrigando o caso de uso a incluir a situação do outro. A inclusão é desenhada por uma linha tracejada com uma seta em uma das extremidades, já as associações de extensão descrevem cenários opcionais entre casos de uso. A extensão é representada por uma linha tracejada com uma seta em uma extremidade apontando para o caso de uso que extende. A Figura 2 exemplifica as associações de casos de uso.



**Figura 2: Exemplo de Casos de Uso**

Fonte: Grupo CRPO(2015)

Cada caso de uso presente no diagrama é associado a uma tabela de documentação. Esta documentação descreve o caso de uso por meio de uma linguagem simples. Para exemplificar o padrão adotado neste trabalho segue a Tabela 2. Os atores relacionados, as condições de início e término, a sequência de passos do caso de uso e as exceções que podem ocorrer são descritas abaixo.

**Tabela 2: Modelo de Documentação do Caso de Uso**

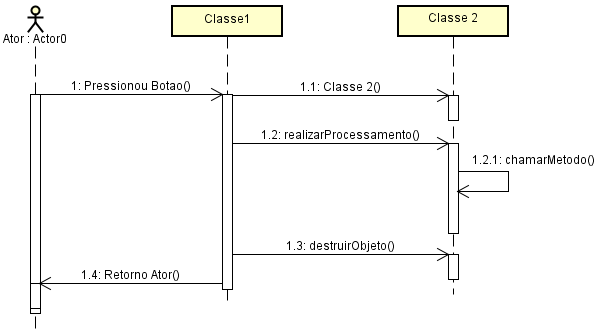
|  |  |
| --- | --- |
| **Caso de Uso** | Nome do caso de uso |
| **Descrição** | Descritivo do caso de uso. |
| **Atores** | O ator do caso de uso. |
| **Pré-Condição** | Condição inicial do sistema para que o caso de uso seja executado. |
| **Pós-Condição** | Estado final do sistema após a execução do caso de uso. |
| **Requisitos Associados** | Identificador e nome do requisito associado, este requisito deve ser apresentado na seção de requisitos do sistema. |
| **Fluxo Principal:** | |
| 1 Ação do sistema | |
| **Tratamento das Exceções:** | |
| 1.a Ocorreu uma exceção durante a execução da ação do sistema | |

Fonte: Grupo CRPO(2016)

**1.3 DIAGRAMAS DE SEQUÊNCIA**

É um diagrama comportamental que determina a sequência de eventos que ocorrem em um determinado processo. Ele identifica também a ordem das mensagens que devem ser enviadas em relação ao tempo. O diagrama de sequência é compostos por classes e por um ou mais atores (associados ao diagrama de casos de uso), que solicita um serviço e inicia os eventos em um processo *(UML 2, GUEDES, 2011).*

Cada componente do diagrama possui uma *lifetime* que representa o tempo que um objeto existe em um processo, desenhado por uma linha fina tracejada na vertical. Um objeto está interagindo quando ele possui um foco de controle ou ativação apresentado dentro da *lifetime* por uma linha mais grossa. O objeto pode realizar chamadas aos seus próprios métodos está chama é representada por uma seta que volta para o mesmo objeto e para a comunicação entre os objetos e atores do diagramas existem as mensagens, elas demonstram a ocorrência de um evento e forçam a chamada de um método. A mensagem é representada por uma seta que liga os dois componentes. Caso algum método seja chamado este vai pertencer ao objeto que recebe a seta. Os métodos ou eventos podem possuir retornos representados por uma seta tracejada que indica a direção do mesmo. É possível ainda neste diagrama construir e destruir objetos *(UML 2, GUEDES, 2011).* A Figura 3 apresenta um exemplo de diagrama de sequência com os componentes citados.



**Figura 3: Exemplo de Diagrama de Sequência**

Fonte: Grupo CRPO(2016)

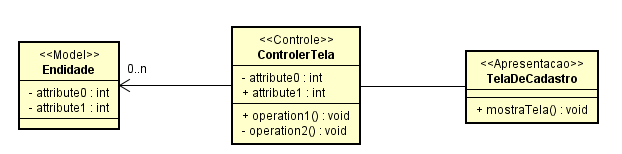
**1.4 DIAGRAMAS DE CLASSE**

Serve para visualizar as classes que compõem o sistema e os seus respectivos atributos e métodos bem como o relacionamento e transmissão de informações entre as classes. A principal preocupação do diagrama de classes é definir a estrutura lógica do sistema *(UML 2, GUEDES, 2011).*

A classe possui três divisões para o nome da classe, atributos e métodos. A visibilidade pública é representada por meio de um sinal de mais(+) o atributo e para o método. Quando o atributo for privado da classe, ou seja, somente acessível dentro da própria classe será representado sinal de menos(-). Os métodos *get* e *set a*tributos privados das classes serão omitidos no diagrama de classes, mas estarão presentes na implementação.

As classe poderão possuir um estereótipos que indica que a classe realiza uma função diferente das demais. Para este projeto serão utilizados três estereótipos conforme descrito anteriormente. O *Apresentação* indica que a classe realiza a comunicação entre os atores e o sistema. O *Controle* faz o intermédio entre as classes *Apresentação* e as demais do sistema. O *Model* torna explícito que a classe é uma entidade, ou seja, contém as informações recebidas ou geradas pelo sistema.

Outro elemento importante do diagrama de classes são as associações que conectam as classes. A associação binária indica o relacionamento entre duas classes a seta indica o sentido que as informações são transmitidas, sem a flecha as informações trafegam em ambos os sentidos. Nas extremidades da reta que liga as duas classes existem as multiplicidades que indicam o número máximo e mínimo de objeto envolvidos na associação, quando este valor for omitido entende-se que significa “1.1”.



**Figura 4: Exemplo de Diagrama de Classes**

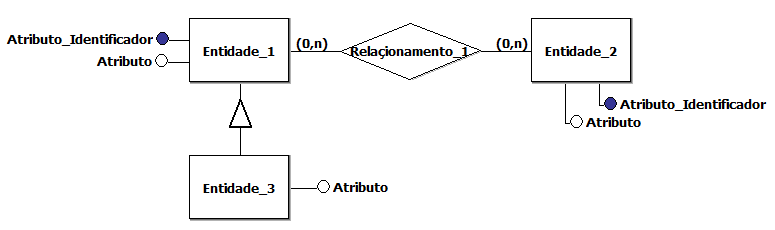
Fonte: Grupo CRPO(2016)

**1.5 DIAGRAMA ENTIDADE-RELACIONAMENTO**

Fornece uma descrição abstrata, dos dados que serão armazenados no banco de dados, pertencendo ao modelo conceitual. O modelo conceitual registra os dados que podem ser gravados no banco de dados, mas não especifica como estes dados serão armazenados no banco.

O diagrama de ER é composto basicamente de entidades e relacionamentos. O conjunto de objetos da realidade modelada sobre os quais deseja-se manter informações no banco de dados, são as entidades. Elas são representadas graficamente com um retângulo que possui o nome da entidade. Os relacionamentos tem por definição como conjunto de associações entre entidades. Os relacionamentos são representados por um losango com o nome do mesmo. Cada relacionamento possui um certo número de associações com outra entidade (cardinalidade), tendo um valor mínimo e um valor máximo.

As entidades e os relacionamentos podem possuir atributos. Os atributos representam os dados da ocorrência. São representados graficamente por um círculo conectado a caixa de entidade ou relacionamento. Quando este atributo for representado por um círculo preenchido na cor preta significa que este é um atributo identificador. O conceito de herança é aplicado também aos diagramas ER, representado por um triangulo entre a classe pais e filha.



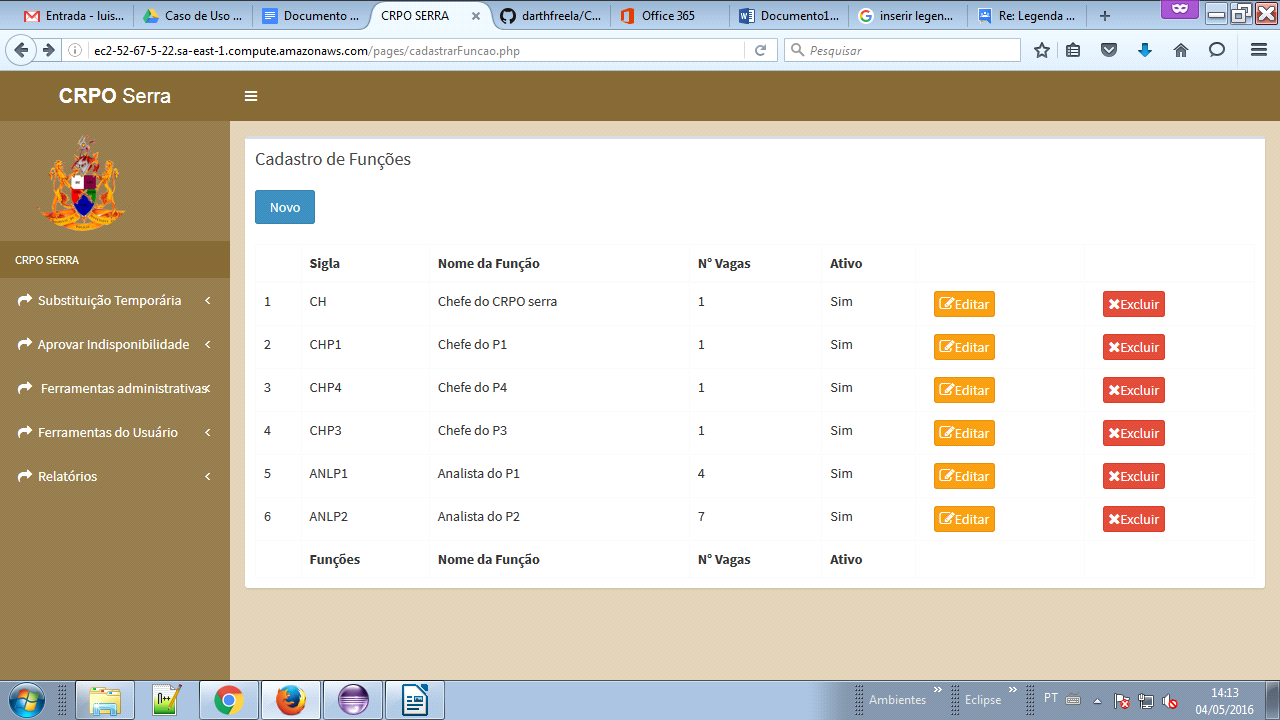
**Figura 5: Exemplo de diagrama ER**

Fonte: Autor (2016)

**1.6 PROTÓTIPO DE TELA**

As telas foram prototipadas em como modelagem por meio de código e Html e folhas de estilos em Css. Já visando uma melhor visualização do cliente como será o sistema, possibilitando mais rapidez em futuras mudanças.

A Figura 6 apresenta um modelo de tela construída com algumas listagem e componentes para o uso.

**Figura 6: Exemplo de Tela Construída**

Fonte: Grupo CRPO(2016)